

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN
APPLICATION DATA SHEET (37 CFR 1.76)**

As the below named inventor(s), I/we declare that:

This declaration is directed to:

- ☒ The attached application, or
☐ Application No. _____, filed on _____
☐ as amended on _____ (if applicable);

I/we believe that I/we am/are the original and first inventor(s) of the subject matter which is claimed and for which a patent is sought;

I/ we have reviewed and understand the contents of the above-identified application, including the claims, as amended by any amendment specifically referred to above;

I/we acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me/us to be material to patentability as defined in 37 CFR 1.56, including material information which became available between the filing date of the prior application and the National or PCT International filing date of the continuation-in-part application, if applicable; and

All statements made herein of my/own knowledge are true, all statements made herein on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001, and may jeopardize the validity of the application or any patent issuing thereon.

FULL NAME OF INVENTOR(S)

Inventor one: Bruce D. Melick Date: 1-31-2001

Signature: Bruce Melick Citizen of: USA

Inventor two: David M. Snyder Date: 1-31-2001

Signature: D. M. Snyder Citizen of: USA

Inventor three: Leslie D. Baych Date: 1-31-01

Signature: Leslie D. Baych Citizen of: USA

Inventor four: _____ Date: _____

Signature: _____ Citizen of: _____

☐ Additional inventors are being named on _____ additional form(s) attached hereto.

Burden Hour Statement: This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is used by the public to file (and the PTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This form is estimated to take 1 minute to complete. This time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

ATTORNEY DOCKET: P04879US1

Please type a plus sign (+) inside this box → ☐

PTO/SB/81 (10-00)

Approved for use through 10/31/2002. OMB 0651-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY OR AUTHORIZATION OF AGENT

Application Number	TBA
Filing Date	
First Named Inventor	MELICK et al.
Group Art Unit	TBA
Examiner Name	TBA
Attorney Docket Number	P04879US1

I hereby appoint:

☒ Practitioners at Customer Number

22885

Place Customer
Number Bar Code
Label here

☐ Practitioner(s) named below:

Name	Registration Number

as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Please change the correspondence address for the above-identified application to:

☒ The above-mentioned Customer Number.

OR

☐ Firm or
Individual Name

Address

Address

City

State

Zip

Country

Telephone

Fax

I am the:

☒ Applicant/Inventor.

☐ Assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

SIGNATURE of Applicant or Assignee of Record

Name

Bruce D. Melick

Signature

Bruce Melick

Date

1.31.2001

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☒ Total of 3 forms are submitted.

Burden Hour Statement: This form is estimated to take 3 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231

Please type a plus sign (+) inside this box → ☐

PTO/SB/81 (10-00)

Approved for use through 10/31/2002. OMB 0651-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY OR AUTHORIZATION OF AGENT

Application Number	TBA
Filing Date	
First Named Inventor	MELICK et al.
Group Art Unit	TBA
Examiner Name	TBA
Attorney Docket Number	P04879US1

I hereby appoint:

☒ Practitioners at Customer Number

22885

Place Customer
Number Bar Code
Label here

☐ Practitioner(s) named below:

Name	Registration Number

as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Please change the correspondence address for the above-identified application to:

☒ The above-mentioned Customer Number.

OR

☐ Firm or
Individual Name

Address

Address

City

State

Zip

Country

Telephone

Fax

I am the:

☒ Applicant/Inventor.

☐ Assignee of record of the entire interest. See 37 CFR 3.71.

Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

SIGNATURE of Applicant or Assignee of Record

Name Leslie D. Baych

Signature

Date

1.31.01

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☒ Total of 3 forms are submitted.

Burden Hour Statement: This form is estimated to take 3 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

Please type a plus sign (+) inside this box → +

PTO/SB/81 (10-00)

Approved for use through 10/31/2002. OMB 0851-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY OR AUTHORIZATION OF AGENT

Application Number	TBA
Filing Date	
First Named Inventor	MELICK et al.
Group Art Unit	TBA
Examiner Name	TBA
Attorney Docket Number	P04879US1

I hereby appoint:

☒ Practitioners at Customer Number

22885

Place Customer
Number Bar Code
Label here

OR
☐ Practitioner(s) named below:

Name	Registration Number

as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Please change the correspondence address for the above-identified application to:

☒ The above-mentioned Customer Number.

OR

☐ Firm or
Individual Name

Address

Address

City

State

Zip

Country

Telephone

Fax

I am the:

☒ Applicant/Inventor.

☐ Assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

SIGNATURE of Applicant or Assignee of Record

Name David M. Snyder

Signature

Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☒ Total of 3 forms are submitted

Burden Hour Statement: This form is estimated to take 3 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

method open(var eventInfo Event)

```

; 1/12/97
; Bruce Melick - OnTheEdge Solutions
;
; =====
; ELECTRONIC SYSTEM FOR RECORDING, STORING, AND RETREIVING MATERIAL HANDLING EQUIPMENT RECORDS AND CERTIFICATIONS
; =====

if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form:
else
    ;// This code executes only for the form:
doDefault

; attempt to open a tcursor on the network....

If NOT tc.open("SHARETBL:ITEMMSTR.DB") THEN
    msgStop("Stop","Cannot locate tables.\nThis application requires the use of network tables.\n\nAre you logged into the network properly?")
    exit()
ELSE
    hideshowbar()
    tc.close()
    mm.addtext("&Chain")
    mm.addtext("&WireRope")
    mm.addtext("&Mesh")
    mm.addtext("&Web")
    mm.addtext("&Round")
    mm.addtext("&Exit")
    mm.show()
ENDIF
endif
endmethod

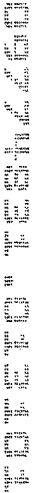
```

```
method menuAction(var eventInfo MenuEvent)
var
  mc      String
endvar

if eventInfo.isPreFilter() then
  // This code executes for each object on the form:
else
  // This code executes only for the form:
  mc = eventInfo.MenuChoice()

  SWITCH
    CASE mc = "&Chain":
      pushButton(btnChain)
    CASE mc = "&WireRope":
      pushButton(btnWireRope)
    CASE mc = "&Mesh":
      pushButton(btnMesh)
    CASE mc = "&Web":
      pushButton(btnWeb)
    CASE mc = "&Round":
      pushButton(btnRound)
    CASE mc = "&Exit":
      pushButton(btnClose)
  ENDSWITCH

  endif
endmethod
```



```
method pushButton(var eventInfo Event)
; open a form that allows user to work with chain sling data
IF f.open("mmChain") THEN
    f.wait()
ELSE
    MsgStop("STOP","Cannot find input form")
ENDIF
endmethod
```

=====

method open(var eventInfo Event)

; Bruce Melick - OnTheEdge Solutions

;

;

;

; ELECTRONIC SYSTEM FOR RECORDING, STORING, AND RETREIVING MATERIAL HANDLING EQUIPMENT RECORDS AND CERTIFICATIONS

;

=====

if eventInfo.isPreFilter() then
 ; // This code executes for each object on the form:

else

 ; // This code executes only for the form:

doDefault

; attempt to open a tcursor on the network....

If NOT tc.open("":SLINGS:SLINGS.DB") THEN

 msgStop("Stop", "Cannot locate tables.\nThis application requires the use of network tables.\n\nAre you logged into the network properly?")
 exit()

ELSE

 hidespeedbar()

 tc.close()

 mm.addtext("Input")

 mm.addtext("Inspect")

 mm.addtext("Reports")

 mm.addtext("Return")

 mm.show()

ENDIF

endif

endmethod

method RptView()

```
Var
  RptToolBar  Form
endVar
```

```
if RptToolBar.Open(":SHAREFRM:TOOLBAR8.FDL") then
  RptToolBar.setPosition(100,650,5600,950)
  RptToolBar.DisplayReport(vReportName)
  RptToolBar.Wait()
  RptToolBar.Close()
endif
```

endmethod

```
method menuAction(var eventInfo MenuEvent)
var
  mc      String
endvar

if eventInfo.isPreFilter() then
  // This code executes for each object on the form:
else
  // This code executes only for the form:
  mc = eventInfo.MenuChoice()

  SWITCH
    CASE mc = "Input":
      pushButton(btnInput)
    CASE mc = "Inspect":
      pushButton(btnInspect)
    CASE mc = "Reports":
      pushButton(ReportsBtn)
    CASE mc = "Return":
      pushButton(CloseBtn)
  ENDSWITCH

  endif
endmethod
```

```
method arrive(var eventInfo MoveEvent)

if eventInfo.isPreFilter() then
    // This code executes for each object on the form:
else
    // This code executes only for the form:
    ; put up a message dialog box for future use...

    MessageDlg.open(":SHAREFRM:MSGDLG8.FDL",WinStyleHidden)
    MessageDlg.setPosition(4500,4000,4000,2350)

    ; -----
    ; take a copy of various tables for use later
    ; -----

    destTbl = ":PRIV:params.db"
    srcTbl.attach(":sharetbl:params.db")
    srcTbl.copy(destTbl)
    srcTbl.unattach()

    ;-----
    destTbl = ":slings:LinkMfg.db"
    srcTbl.attach(":slings:Mfg.db")
    srcTbl.copy(destTbl)
    srcTbl.unattach()

    ;-----
    destTbl = ":slings:LinkSize.db"
    srcTbl.attach(":slings:Size.db")
    srcTbl.copy(destTbl)
    srcTbl.unattach()

endif

endmethod
```

```
method open(var eventInfo Event)

if eventInfo.isPreFilter() then
    // This code executes for each object on the form
else
    // This code executes only for the form
    doDefault
    action(DataBeginEdit)
    IF NOT lib.open(":sharelib:varlib1.lsl",GlobalToDesktop) THEN
        Message("Could not open library")
        return
    ENDIF
    mm.addtext("")
    mm.show()
endif
endMethod
```

Page 1: slinginput::#Formdata1::arrive

```
method arrive(var eventInfo MoveEvent)
  if eventInfo.isPreFilter() then
    // This code executes for each object on the form
  else
    // This code executes only for the form
    MessageDlg.open("sharefrm:MSGDLG8.FDL",WinStyleHidden)
    MessageDlg.setPosition(4500,4000,4000,2350)
  endif
endMethod
```

Page 1: inspect::#Formdata1::open

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form:
  else
    ;// This code executes only for the form:
    ; set the delete flag default..
    delflag = "nogo"
    doDefault

    IF NOT lib.open("sharelib:varlib1.lsl",GlobalToDesktop) THEN
      Message("Could not open library")
      return
    ENDIF

    mm.addtext("")
    mm.show()
    action(dataBeginEdit)
    tbl1.attach(tblSlings)
  endif
endmethod
```

Page 1: inspect::#Formdata1::keyPhysical

method keyPhysical(var eventInfo KeyEvent)

```
if eventInfo.isPreFilter() then
    // This code executes for each object on the form:
    ; trap any users key strokes...this is inquiry only!
```

SWITCH

```
CASE eventInfo.vChar() = "VK_F6" AND pgRecord.focus = true :
    ; function key for moving to page two
    disableDefault
```

moveToPage(2)

```
CASE eventInfo.vChar() = "VK_F6" AND pgDetail.focus = true :
    ; function key for moving to page one
    disableDefault
```

moveToPage(1)

```
CASE eventInfo.vChar() = "VK_F7" :
    MsgInfo("Note","You cannot view the underlying tables from this input form.")
    disableDefault
```

```
CASE eventInfo.vChar() = "VK_INSERT" AND pgRecord.focus = true :
    MsgInfo("Note","You cannot insert new sling records from this form.")
    disableDefault
```

```
CASE eventInfo.vChar() = "VK_INSERT" AND pgDetail.focus = true :
    disableDefault
```

pushButton(InsertRecButton)

```
CASE eventInfo.vChar() = "VK_DELETE" AND pgRecord.focus = true :
    MsgInfo("Note","You cannot delete sling records from this form.")
    disableDefault
```

disableDefault

```
CASE eventInfo.vChar() = "VK_DELETE" AND pgDetail.focus = true :
    disableDefault
    pushButton(DeleteButton)
```

ENDSWITCH

else

```
    ;// This code executes only for the form:
```

endif

endmethod

Formdata1::DelAll

method DelAll()

; delete the master equipment record and ALL detail inspection records
; associated with the equipment

```
var
  DelChoice2,delno      String
  TblObj                UIObject
  delcount              Number
endvar
```

; the [Del] button is pushed.

; Delete the detail records first.....

```
delno = SlingID.value
DelChoice2 = msgQuestion("FINAL WARNING","You will delete a Sling Record: "+STRVAL(delno)+" and ALL of the associated inspection records!?\n
IF DelChoice2 = "No" THEN
  RETURN
ENDIF
```

; open a tcursor on maint.db and delete all the details records first....

```
tcl.open("":slings:maint.db")
tcl.edit()
delcount = 0
```

```
IF tcl.locate("SlingID",delno) THEN
  tcl.deleteRecord()
  delcount = delcount+1
  tcl.home()
  while tcl.locateNext("SlingID",delno)
    tcl.deleteRecord()
    delcount = delcount+1
    tcl.home()
  endwhile
ENDIF
```

```
tcl.endEdit()
tcl.close()
```

; now that all the detail records are gone, delete the master....
action(DataDeleteRecord) ; deletes the current record on the screen

IF delcount <> 0 THEN

```
  msgInfo("Records Deleted","Sling ID Record: "+STRVAL(FORMAT("w6",delno))+" and "+STRVAL(FORMAT("w6",delcount))+" detail inspection records
ENDIF
```

endmethod

Page 1: inspect::#Formdata1::arrive

```
method arrive(var eventInfo MoveEvent)
  if eventInfo.isPreFilter() then
    // This code executes for each object on the form:
  else
    // This code executes only for the form:
    MessageDlg.open("sharefrm:MSGDLG8.FDL",WinStyleHidden)
    MessageDlg.setPosition(4500,4000,4000,2350)
    action(dataBeginEdit)
  endif
endmethod
```

inspect::#Formdata1::action

method action(var eventInfo ActionEvent)

if eventInfo.isPreFilter() then
 // This code executes for each object on the form:

; if user presses Ctrl|Del then stop them from proceeding
; the delete flag (delflag) must be set to "go" and the only
; way that can happen is if the user presses the DEL button

IF eventInfo.id() = dataDeleteRecord AND delflag = "nogo" THEN
 msgInfo("STOP", "To delete sling inspection records, use the [DEL] Button.")
 disableDefault
ENDIF

else
 // This code executes only for the form:

endif

endmethod

```
method arrive(var eventInfo MoveEvent)
```

```

// This code executes for each object on the form

```

```
method open(var eventInfo Event)
```

```
  if eventInfo.isPreFilter() then
```

```
    // This code executes for each object on the form
```

```
  else
```

```
    // This code executes only for the form
```

```
  doDefault
```

```
    action(DataBeginEdit)
```

```
    IF NOT lib.open(":sharelib:varlib1.lsl",GlobalToDesktop) THEN
```

```
      Message("Could not open library")
```

```
      return
```

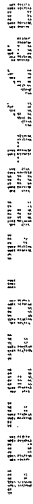
```
    ENDIF
```

```
    mm.addtext("")
```

```
    mm.show()
```

```
  endIf
```

```
endMethod
```



```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    /// This code executes for each object on the form
  else
    /// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the BRANCHES.DB table...
  tc.open(":slings:branches.db")
  endif
endMethod
```

```
method arrive(var eventInfo MoveEvent)
    if eventInfo.isPreFilter() then
        ;// This code executes for each object on the form
    else
        ;// This code executes only for the form
        ; using the variable passed from the calling form (current code field value)
        ; locate the code in the courses table

        incoming = lib.getVar() ;take the variable passed from calling form
        ;incoming.view("Receiving this from the library...")
        ; incoming.view()

        ; find manufacturer name in table
        if tc.locate("Branches", incoming) then
            tblBranches.resync(tc)
        endif
        tblBranches.moveTo()
    endif
endMethod
```

Page 1: listcust::#Formdata1::Open

```
method open(var eventInfo Event)

if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
else
    ;// This code executes only for the form
doDefault
IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
ENDIF
; open a tcurser on the Customer Master table...
tc.open(":slings:custmast.db")

endif
endMethod
```

```
method arrive(var eventInfo MoveEvent)
if eventInfo.isPreFilter() then
    ;;; This code executes for each object on the form
else
    ;;; This code executes only for the form
; using the variable passed from the calling form (current code field value)
; locate the code in the courses table

incoming = lib.getVar() ;take the variable passed from calling form
incoming.view("Receiving this from the library...")
; incoming.view()

; find manufacturer name in table
if tc.locate("CUST_NO", incoming) then
    tblCust.resync(tc)
endif
tblCust.moveTo()

endif
endMethod
```

Page 1: listdamage::#Formdata1::open

Formdata1::open

```
method open(var eventInfo Event)
if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
else
    ;// This code executes only for the form
doDefault
IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
ENDIF
; open a tcursor on the INSPECTOR.DB table...
tc.open(":slings:damage.db")
endif
endMethod
```

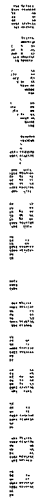
Formdata1::arrive

```
method arrive(var eventInfo MoveEvent)
    if eventInfo.isPreFilter() then
        ;// This code executes for each object on the form
    else
        ;// This code executes only for the form
        ; using the variable passed from the calling form (current code field value)
        ; locate the code in the courses table

        incoming = lib.getVar() ;take the variable passed from calling form
        ;incoming.view("Receiving this from the library...")
        ; incoming.view()

        ; find manufacturer name in table
        if tc.locate("Damage", incoming) then
            tblDamage.resync(tc)
        endif
        tblDamage.moveTo()

    endif
endMethod
```



```
method open(var eventInfo Event)

if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
else
    ;// This code executes only for the form
doDefault
IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
ENDIF
; open a tcurser on the GRADE.DB table...
tc.open(":slings:grades.db")
endif
endMethod
```

endIf

Page 1: listhook::#Formdata1::open

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the TYPE.DB table...
  tc.open(":slings:hook.db")
  endIf
endMethod
```

[illegible]

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    /// This code executes for each object on the form
  else
    /// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the INSPECTOR.DB table...
  tc.open(":slings:inspector.db")
  endif
endMethod
```



```
method open(var eventInfo Event)
    if eventInfo.isPreFilter() then
        /// This code executes for each object on the form
    else
        /// This code executes only for the form
    doDefault
    IF NOT lib.open(":\sharelib\varlib1.lsl") THEN
        Message("Could not open library")
        return
    ENDIF
    ; open a tcursor on the TYPE.DB table...
    tc.open(":\slings:link.db")
    endIf
endMethod
```

[illegible]

Page 1: listmfg:~#Formdata1::open

method open(var eventInfo Event)

if eventInfo.isPrefilter() then
 ; // This code executes for each object on the form

else

 ; // This code executes only for the form

doDefault

IF NOT lib.open(":sharelib:varlib1.lsl") THEN

 Message("Could not open library")

 return

ENDIF

 ; open a tcursor on the MFG.DB table...

tc.open(":slings:mfg.db")

endIf

endMethod

Formdata1: arrive

```
method arrive(var eventInfo MoveEvent)
    if eventInfo.isPreFilter() then
        ;// This code executes for each object on the form
    else
        ;// This code executes only for the form
        ; using the variable passed from the calling form (current code field value)
        ; locate the code in the courses table

        incoming = lib.getVar() ;take the variable passed from calling form
        ;incoming.view("Receiving this from the library...")
        ; incoming.view()

        ; find manufacturer name in table
        if tc.locate("Mfg", incoming) then
            MfgTable.resync(tc)
        endif
        MfgTable.moveTo()
    endif
endMethod
```

Page 1: listother::#Formdata1::open

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tursor on the OTHER.DB table...
  tc.open(":slings:other.db")
  endIf
endMethod
```

Variable	Mean	SD	Min	Max
Age	34.5	10.2	18	65
Gender	0.5	0.5	0	1
Marital status	0.6	0.5	0	1
Education	12.5	1.5	9	16
Income	1500	500	500	3000
Health status	0.8	0.2	0	1
Work status	0.7	0.3	0	1
Life satisfaction	4.5	1.0	1	7
Depression	0.2	0.4	0	1
Stress	3.5	1.5	1	7
Resilience	5.5	1.0	3	7
Optimism	5.0	1.0	3	7
Gratitude	5.5	1.0	3	7
Forgiveness	5.0	1.0	3	7
Empathy	5.5	1.0	3	7
Compassion	5.0	1.0	3	7
Kindness	5.5	1.0	3	7
Generosity	5.0	1.0	3	7
Patience	5.5	1.0	3	7
Self-control	5.0	1.0	3	7
Emotional stability	5.5	1.0	3	7
Interpersonal skills	5.0	1.0	3	7
Problem-solving skills	5.5	1.0	3	7
Decision-making skills	5.0	1.0	3	7
Communication skills	5.5	1.0	3	7
Leadership skills	5.0	1.0	3	7
Teamwork skills	5.5	1.0	3	7
Conflict resolution skills	5.0	1.0	3	7
Stress management skills	5.5	1.0	3	7
Emotional regulation skills	5.0	1.0	3	7
Self-awareness	5.5	1.0	3	7
Empathy	5.0	1.0	3	7
Compassion	5.5	1.0	3	7
Kindness	5.0	1.0	3	7
Generosity	5.5	1.0	3	7
Patience	5.0	1.0	3	7
Self-control	5.5	1.0	3	7
Emotional stability	5.0	1.0	3	7
Interpersonal skills	5.5	1.0	3	7
Problem-solving skills	5.0	1.0	3	7
Decision-making skills	5.5	1.0	3	7
Communication skills	5.0	1.0	3	7
Leadership skills	5.5	1.0	3	7
Teamwork skills	5.0	1.0	3	7
Conflict resolution skills	5.5	1.0	3	7
Stress management skills	5.0	1.0	3	7
Emotional regulation skills	5.5	1.0	3	7
Self-awareness	5.0	1.0	3	7

```
method open(var eventInfo Event)
  if eventInfo.isPrefilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the SIZE.DB table...
  tc.open(":slings:linksize.db")
  endif
endMethod
```

Parameter	Value	Unit	Source
Age	1.2	yr	1
Weight	70	kg	1
Height	1.75	m	1
Sex	Male		1
Activity	Low		1
Temperature	37.0	°C	1
Heart rate	70	beats/min	1
Blood pressure	120/80	mmHg	1
Respiratory rate	12	breaths/min	1
Oxygen saturation	98	%	1
Glucose	100	mg/dL	1
Urea nitrogen	10	mg/dL	1
Creatinine	1.0	mg/dL	1
Calcium	9.0	mg/dL	1
Phosphorus	3.0	mg/dL	1
Potassium	4.0	mg/dL	1
Sodium	140	mg/dL	1
Chloride	100	mg/dL	1
Bicarbonate	24	mg/dL	1
Lactate	1.0	mg/dL	1
Ammonia	10	mg/dL	1
Bilirubin	1.0	mg/dL	1
Albumin	4.0	g/dL	1
Protein	7.0	g/dL	1
Triglycerides	150	mg/dL	1
Cholesterol	200	mg/dL	1
Hemoglobin	15	g/dL	1
Hematocrit	45	%	1
Red blood cells	4.5	million/mm ³	1
White blood cells	10	thousand/mm ³	1
Platelets	150	thousand/mm ³	1
Prothrombin time	12	sec	1
Partial thromboplastin time	30	sec	1
Fibrinogen	300	mg/dL	1
D-dimer	0.5	mg/dL	1
C-reactive protein	1.0	mg/dL	1
Erythrocyte sedimentation rate	10	mm/hr	1
Uric acid	5.0	mg/dL	1
Urea nitrogen	10	mg/dL	1
Creatinine	1.0	mg/dL	1
Calcium	9.0	mg/dL	1
Phosphorus	3.0	mg/dL	1
Potassium	4.0	mg/dL	1
Sodium	140	mg/dL	1
Chloride	100	mg/dL	1
Bicarbonate	24	mg/dL	1
Lactate	1.0	mg/dL	1
Ammonia	10	mg/dL	1
Bilirubin	1.0	mg/dL	1
Albumin	4.0	g/dL	1
Protein	7.0	g/dL	1
Triglycerides	150	mg/dL	1
Cholesterol	200	mg/dL	1
Hemoglobin	15	g/dL	1
Hematocrit	45	%	1
Red blood cells	4.5	million/mm ³	1
White blood cells	10	thousand/mm ³	1
Platelets	150	thousand/mm ³	1
Prothrombin time	12	sec	1
Partial thromboplastin time	30	sec	1
Fibrinogen	300	mg/dL	1
D-dimer	0.5	mg/dL	1
C-reactive protein	1.0	mg/dL	1
Erythrocyte sedimentation rate	10	mm/hr	1
Uric acid	5.0	mg/dL	1

Page 1: listtype::#Formdata1::open

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the TYPE.DB table...
  tc.open(":slings:type.db")
  endif
endMethod
```

```
method arrive(var eventInfo MoveEvent)
    if eventInfo.isPreFilter() then
        ;// This code executes for each object on the form
    else
        ;// This code executes only for the form
    ; using the variable passed from the calling form (current code field value)
    ; locate the code in the courses table

    incoming = lib.getVar() ;take the variable passed from calling form
    ;incoming.view("Receiving this from the library...")
    ; incoming.view()

    ; find manufacturer name in table
    if tc.locate("Type", incoming) then
        tblType.resync(tc)
    endif
    tblType.moveTo()

    endif
endMethod
```

,